Computing News

News from the Computing Division

Fermi National Accelerator Laboratory

April 1997

Table of Contents

Welcome to the first online-only issue

Welcome to the first edition of the Fermilab Computing Division Newsletter to be published on line rather than being mailed to our newsletter mailing list. Doing this allows us to deliver the newsletter to those who want it and only to those who want it. It also saves paper, time, and money over delivering a paper copy to several thousand people plus the time needed to maintain the newsletter mailing list. As resources become tighter, this is one way to do more with less. It also gives us the potential for a more useful newsletter because we can link to additional material directly, use color when appropriate, and we are not limited by the constraints of a certain number of 8 by 11 inch paper sheets.

If you want to be notified when new newsletters are released, you can subscribe to the **`cdnews`** mailing list. To do this, send email to **`mailserv@fnal.gov`**. The subject field should be left blank. (Quickmail users must use the Plain Memo form.) The text of the message should contain **`subscribe cdnews`**. Please do not append your signature file for mailserv commands.

Judith Nicholls, editor, nicholls@fnal.gov, x3989
http://www.fnal.gov/cd/CDN/index.html

Computing Division Documentation

Some of you visiting us on the 8th floor may have noticed there is no longer a large room devoted to handing out paper copies of documents and housing reference materials. As more and more documents became available on line, the stacks of paper became more and more irrelevant.

Technology has changed so that it is now relatively easy for us to make information available to everyone on line. The use of the World Wide Web empowers everyone with a computer to avail themselves of a free “browser” with which they can view our materials.

The rapid changes in technology result in rapid turnover in popular documents and rapid obsolescence of documents, which resulted in high labor, duplicating, and paper costs. With our documents on line, we can update them as needed, and you can view

or print them as needed. You always get the latest information by viewing on line and we don't have to worry about processes to move documents into the library, duplicate them, and throw away obsolete copies.

We are still providing copies of some of the documents, mostly large ones that are difficult to view and print, available 24 hours a day outside of the Wilson Hall 8NE. Just inside the door is an X terminal which you can use to view and print the documents. Yolanda Valadez is also still available in that room for account management, password changes, resource requests, etc.

The most popular and up-to-date documents are already on line and available to you using a browser. To find our documentation, go to the Documentation and Software section on the Computing Division home page and select Documentation. Either enter a search keyword (product name, for example) and press search button or select categories below to get a list of available software products or documents. If that isn't successful, try the other resources at the bottom of the page or other choices under Documentation and Software.

If documents that you need for products or services that we support are missing, please let us know and we will attempt to make them available. We would also appreciate comments on how we can make it easier for you to find the documents or information that you need.

Judy Nicholls, nicholls@fnal.gov

Changing fnalu.fnal.gov

Currently the name fnalu.fnal.gov is an alias for the node **`fibi01.fnal.gov`**. This is a poor choice for the default node and will be changed on May 5, 1997, to the node **`fsui02.fnal.gov`**. This is a change in operating system type from AIX (IBM) to SunOS (Sun). Please read the article FNALU Status and Plans for more details.

Dane Skow, dane@fnal.gov
Lisa Giacchetti, lisa@fnal.gov

FNALU Status and Plans

With the retirement of the FNALV cluster and the VMS migration more generally, usage on the FNALU cluster has increased in both CPU utilization and breadth of types of computing jobs. Users have experienced some inconveniences as we have reorganized the cluster to accommodate these changes. This article and the several accompanying articles in this newsletter describe our current situation and plans for the cluster. Hopefully these will give you some information about how to utilize the central Unix cluster effectively and how to avoid some of the more common pitfalls.

First, what is meant by the phrase "FNALU cluster"? This cluster is a group of 17 Unix computers that share a common user registration (password system) and file system (so that your login directory is available from all machines). The machines include nodes from all four of the currently supported Unix vendors (DEC, IBM, SGI, Sun) and one of the major uses of the cluster is as the primary cross development facility for the lab. While the operating system (Unix) is similar across all the machines, the executable files are not compatible among the various vendors. Furthermore, there are differences among the vendors and their implementations of the compilers, for example, that mean that one needs to be aware of the machine you are using. This is one striking difference from other implementations of computing clusters (VMS and single vendor Unix) around the laboratory. For some users this is a big advantage (you have access to all flavors for verifying code for distribution, easily accessing the excess computational power of another Unix flavor, etc.), but for others it is nothing but a complication. The advice for the latter users is to pick one flavor of computer and ignore the rest of the cluster. This typically reduces your choice of computers to one of a couple of largely interchangeable systems. The biggest exception to this statement is email where you currently need to choose a particular machine on which to read new mail (foldered mail is available throughout the cluster). There is a separate article in this newsletter devoted to email configuration and use in the FNALU cluster.

The FNALU cluster services both interactive and batch jobs. Batch jobs are typically CPU intensive or production class jobs. Recently, FNALU has absorbed the Clustered Large Unix Batch System (CLUBS) to help make batch services for Fermi Unix users more convenient. Batch services on FNALU and CLUBS are now integrated into a single software subsystem and users no longer need separate accounts to run batch jobs on what were CLUBS hosts. Thus, apart from restricting interactive logins on the former CLUBS hosts, the distinction between CLUBS and FNALU as separate clusters has largely disappeared.

The machines that make up the FNALU cluster are:

TABLE 1.

Name	*Designated Use*	*Machine OS*	*Machine Type*	*Procs*	*Memory*	*Processor Rating*
fibi01	interactive only	AIX3.2.5	RS6000 590	1		62
fsgi01	interactive only	IRIX 5.3	4D/420	1		30
fsui01	interactive only	SunOS 2.5.1	Sparc 20M514	4		38
fibb01	interactive/batch	AIX 3.2.4	RS6000 560	1	192Mb	39
fsgi02	interactive/batch	IRIX 5.3	R4400 Chal-lenge	20	511Mb	84
fsui02	interactive/batch	SunOS 2.5.1	Ultra-167	4	320Mb	93
fdei01	interactive/batch	OSF1 3.2D	2100/A500	4	637Mb	136
fncl00	batch only	AIX 3.2.5	RS6000 370	1	128Mb	49
fncl01	batch only	AIX 3.2.5	RS6000 370	1	128Mb	49
fncl02	batch only	AIX 3.2.5	RS6000 370	1	128Mb	49
fncl04	batch only	IRIX 5.3	R4400 Crimson	1	143Mb	63
fncl05	batch only	IRIX 5.3	R4400 Crimson	1	143Mb	63
fncl06	batch only	IRIX 5.3	R4400 Crimson	1	143Mb	63
fncl07	batch only	IRIX 5.3	R4400 Crimson	1	143Mb	63
fncl09	batch only	AIX 3.2.5	RS6000 370	1	128Mb	49
fncl10	batch only	AIX 3.2.5	RS6000 370	1	128Mb	49
fncl11	batch only	AIX 3.2.5	RS6000 370	1	128Mb	49

The ratings numbers shown above were determined using a program called TINY, a physics Monte Carlo and reconstruction program taken from a Fermilab experiment. These numbers are valid as a rough estimates of the relative machine strengths; actual performance can vary from application to application.

The naming convention for the 4th letter of the machine name was intended to indicate if the machine is primarily for batch or interactive computing (e.g. fibb01 versus fibi01). The convention has largely been ignored in the past but will be used regularly in the future. Towards this end, interactive logins to fibb01 will be disabled in the future when another interactive machine becomes available.

There is a node with an alias fnalu (currently fibi01) to permit connections for novice users who have not yet made their selection of a primary machine. We will change this alias to point to fsui02 on May 5, 1997, to better serve these casual users. Users who require the IBM environment should switch to using the explicit name of one of the IBM machines. In general, we will support the use of the FNALU name for a connection to the machine “best” able to provide general interactive use (email, web browsers, etc.). Which physical machine this is may change from time to time, but we expect this to be no more frequent than quarterly and will try to minimize these disruptions.

Dane Skow, dane@fnal.gov
Lisa Giacchetti, lisa@fnal.gov

AFS File System

One of the differences experienced Unix users see when coming to the FNALU cluster is the use of the AFS file system. The major benefit of this file system is that the same file structure is available from all client machines making for easy exchange of data between different users, systems, and even locations. There are several differences from the usual Unix file system with which a new user must become familiar and they are documented in the Unix at Fermilab chapter devoted to AFS. Most of these are easily accommodated or utilized by the users, others are more fundamental.

What is this AFS system and why is it worth the bothersome stands for the Andrew File System and was originally developed at Carnegie Mellon University as part of the Athena project with IBM. The commercialization and present vendor is Transarc Corporation. The system was designed to create a distributed file system that allowed hundreds or thousands of workstations to share a single view of the file structure. Thus a user could count on accessing the files in **`/afs/fnal.gov/home/user1`** from any workstation or pc that was running AFS. It does not require prior coordination among all the system managers to properly configure each individual workstation and server. In its grandest implementation, this could permit a single login to authenticate and provide common file access from any machine on site.

■

There are several other key features in the AFS design including the use of Kerboros security system for file exchange, server replication for redundant access to data, much more finely controlled file access permissions, and local caching of frequently used data. These features are all used (and useful) to various degrees in the Fermilab environment. A particularly useful example was the recent move of all of the file servers from one floor of Feynman Computing Center to another without the interruption of service (staged moves using the server replication features) something that is not possible with other available file system.

It is the local caching, however, that has given rise to the greatest amount of difficulty for FNAL use (primarily in multiprocessor machines) of the system. One of the major limitations to the AFS implementation is the inability to control the contents of the local AFS cache. There is one cache per machine and one cannot segment it to, for example, protect product libraries and executables from being removed to make way for raw data streaming through some batch data analysis job. Much of the work over the past year has focussed on optimizing the configuration of this cache and working with the vendor to resolve problems uncovered. This continues to be an active area of effort and there have been some significant steps made in the past few months.

In the past, all disk storage on FNALU has utilized the AFS file system. There are in general, 3 different types of user data accessed from the FNALU machines typified by:

TABLE 2.

Source code libraries	User home directories	Data staging areas
Mostly read access	Read/Write access	Mostly stage in and read once
Widely shared	mostly 1 user	Often one user
Access more important than throughput	Both access and throughput important	Throughput most important
Moderate files (<50MB)	Small files (<10MB)	Large files (.1 - 5 GB)

These different types of files have different requirements and some are more appropriate for AFS storage than others, the source code libraries being most ideally suited. However, it is quite clear that we need to relieve some of the data volume through the AFS cache. We are implementing several changes to address that situation. Already several of the nodes in FNALU have 10s of GB of locally installed disk for use as high performance data stores by individual experiments. We are purchasing the components needed to increase that to approximately 100 GB per large machine. We will at the same time install some generally accessible scratch disk on each of the machines (approximately 15 GB) for temporary storage of working files. Since these scratch areas will be prone to filling up, we will implement a cleaning system to prevent this. The time a file remains available in this area will not be guaranteed and none of these areas will be backed up. Users are expected to utilize backed-up AFS volumes for storage of smaller and personal files and the MSS hierarchical storage management system for larger data sets.

We continue to analyze the situation with the user home directories and will determine what appropriate actions to take there after the installation of the local disk is completed (estimated some time in May). All of the currently available alternatives (local disks, NFS servers, DFS servers) have performance concerns as well and would certainly entail significant user disruption while being implemented. Because we understand the impact of the service interruptions experienced in the past 6 months, we are unwilling to introduce more without thorough testing.

Dane Skow, dane@fnal.gov

Lisa Giacchetti, lisa@fnal.gov

Batch computing on FNALU/CLUBS

The central Unix batch computing facility was previously a separate cluster known as CLUBS (Clustered Large Unix Batch System) that was only loosely connected to the FNALU cluster. While batch jobs could be submitted from FNALU, the utilization of CLUBS batch could be complicated by the user accounts and administration of the two clusters being completely separated. For example, one needed to make a special request in order to enable an account to submit a CLUBS batch job. From the user perspective this was further complicated by FNALU having its own batch subsystem.

■

Recently the underlying batch system was changed from IBM's LoadLeveler product to Platform Computing's LSF (Load Sharing Facility). As part of this changeover, the two clusters, CLUBS and FNALU were merged into a single entity and now present a unified batch system to users. While the traditional CLUBS nodes still do not permit interactive logins, they are now available for batch use by all FNALU accounts. Converting from LoadLeveler to LSF has also allowed FNALU to expand batch services onto the FNALU OSF1 and SunOS systems. Computing Division responsibilities have been streamlined with this restructuring as well: the OSS department now is responsible for system administration of both the FNALU and former CLUBS nodes while the HPCC department is responsible for batch software on both FNALU and the former CLUBS nodes.

Job submission into the batch system continues to be through the use of the **fbatch** product. **fbatch** is built as a layer on top of LSF and adds the following functionality:

- isolation for users from minor changes to the underlying batch system
- a user interface to the Fermi **spacall** (space allocator) and **nt** (needfile/tcache) products
- AFS token renewal
- a remote shell batch submission

fbatch was designed to provide some continuity for previous users without reducing any of the underlying functionality of LSF.

LSF creates a single system image of a network of computers. Components of this image are system resources and queues. The user selects the resources and queue required for the job while LSF selects the host most appropriate to run the job. Some examples of resources are:

irix - the job may be executed on any IRIX host

aix - the job may be run on any AIX host

any - the job may by run on any flavor on Unix O/S

Some examples of queues are:

30min - the job can use as most 30 minutes of CPU time as scaled to the most powerful host

12hr_disk - the job can use at most 12 hours of CPU time as scaled to the most powerful host and will be allocated 1.2GB of scratch work space for the duration of the job

e831_short - only user e831tod may submit a job to this queue and the job can use at most 120 minutes of CPU time and run only on fdei01.

LSF dispatches a job from queues based on the priority assigned to the queue. Generally queues with shorter CPU limits have higher priority than queues with longer CPU limits. The host selected by LSF is based on the current load compared to acceptable dispatch thresholds among all eligible hosts. For example, the acceptable dispatch level on an interactive/batch host is set lower than the acceptable dispatch level of a batch only host. LSF can limit the number of batch jobs concurrently executing on a specific host or from a specific queue. Furthermore, LSF can suspend batch jobs whenever the current load exceeds an acceptable peak. Jobs will automatically be resumed on that host when the current load drops back to the acceptable dispatch level.

An overview of the **fbatch** commands, job submission, and LSF is available on the web at http://fnhppc.fnal.gov/fbatch/fbatch.html

As seen in Table 1 above, the computational power available in the FNALU interactive/batch machines is much greater than that of the batch-only machines. This will change as the planned hardware purchases for this fiscal year are focussed on upgrading the dedicated batch computing machines for AIX, IRIX, and OSF/1. In the meantime, we are encouraging users to use the **fbatch** facilities on the existing machines to facilitate their future use of the batch systems and to minimize the impact of long running jobs on the interactive users. The current policy is to permit only two long running background processes per user and to enforce priority lowering via a system monitoring script. The batch system permits a much more flexible load management system and we will likely reduce the permitted number and time of background processes in the future. While significant computational power is available through the interactive/batch systems of FNALU, it will continue to be the case that priority will be given to interactive computing, particularly during normal working hours. We will continue to work with users and develop tools to ensure that this is the case.

Marilyn Schweitzer, marilyn@fnal.gov
Dane Skow, dane@fnal.gov

■

email on FNALU

In this day and age, email (**e**lectronic **mail**) is the lifeline between co-workers and collaborators across the world. Its reliability and ease of use is of great importance and has become a hot topic for users and system administrators alike. Variations between the mail delivery mechanism and the many links in the chain between sender and receiver lead to all sorts of ways in which mail can appear to be lost or not be delivered correctly. It is for these reasons that the Computing Division has tried to establish a set of rules for email configuration to increase the reliability of the service.

The first step was taken several years ago with the installation of a site wide mail server, **`fnal.gov`** (its full name is **`fnal.fnal.gov`**). This system is meant to be the site gateway for all email traffic. Ideally any mail being sent on site or offsite should be processed through this system. Therefore all Fermilab users are given an account on **`fnal.gov`** when they request a computer account. The account is then set up to forward the users' email to the node where they will be reading email regularly. Some Unix systems are even configured to send all outgoing mail through **`fnal.gov`** for processing, bypassing the local mailer entirely. We also encourage all users to send mail via fnal.gov by always using the email address **`<user>@fnal.gov`** for Fermilab users.

The second step to ensure email reliability was taken during the migration off FNALV. Tutorials on the use of supported mail readers were held and an email helpdesk was set up in the atrium at the high rise. During this migration time, users were directed to execute the following steps if they were going to be reading email on a node in FNALU:

1. If the user was moving off FNALV and would be reading mail regularly on FNALU, the user needed to pick one node on FNALU on which they would be reading email.
2. The users forward on **`fnal.gov`** should be set to point to the node picked in step 1.
3. The user's **`.forward`** on all other Unix login areas should be set to point to **`fnal.gov`**.
4. The **`.forward`** in their login area on the Unix cluster where they read their mail (in this case FNALU) should be set to the node within the cluster where they read mail. This node was the one picked in step 1. In this step, the forward should be prepended with a "\" character: **`\lisa@fsui02.fnal.gov`**

This last step needs to be taken because under Unix, mail gets delivered to the system mail spool area of the node specified in the email address the sender uses. So if someone sends me mail using the address **`lisa@dcdlaa.fnal.gov`**, their mail would end up in the **`/var/mail/lisa`** file on dcdlaa.fnal.gov. If **`dcdlaa.fnal.gov`** is part of a "Unix cluster" which has several systems in it and a shared login area for users across it, I would not be able to read the mail sent to dcdlaa from any other node in the cluster even though my login area is shared. This is because the **`/var/mail`** area is local to each node and not shared across the cluster. In investigating complaints about mail problems on FNALU, we have found that about 1100 accounts out of 1600 or so do not have a .forward file in them. This in and of itself could lead to many "lost" messages. For example, suppose that I do not have a .forward file in my account on FNALU and I read all my mail on fsgi02. A user on node dcdsv0 sends me mail using the address **`lisa@fnalu.fnal.gov`** (which is just an alias for a node in fnalu, currently fibi01 and after May 5 fsui02). Their mail would go to the **`/var/mail/lisa`** file on fibi01. This is not currently accessible from fsgi02 where I read my mail so unless I regularly log into fibi01 to read mail, I would never see the message. The same situation could occur if a user on one of the other FNALU nodes sends mail to me without specifying a node name (e.g. **`Mail lisa`**). This mail would stay in the system mail area on the node they sent the mail from. If that node is not fsgi02 and I don't have a forward set, I would never see the mail.

Furthermore, **pine** users do not by default folder their mail once it has been read (the inbox stays in the system area). This means that messages in the **pine** inbox are only available on the specific machine even with a **`.forward`** file properly configured. To make your mail available from any FNALU machine in pine, put your mail in folders.

As of April 7, we will have "retrofitted" every account on FNALU with a **`.forward`** file. There were several steps involved in reaching this goal. The first was to compile a list of all users without a **`.forward`**. We then checked the forward of each of these users on **`fnal.gov`**. If the users forward on **`fnal.gov`** points to a node in FNALU, we set the FNALU login **`.forward`** to point to that node too. If the **`fnal.gov`** forward points to a node not part of FNALU the users **`.forward`** on FNALU will get set to point to **`fnal.gov`**.

The third step being taken to improve mail services on FNALU will be to move the "fnalu" system alias to a different node, fsui02, and then recommend that all users of the FNALU cluster who are basically only there to read email use this node. As stated above, connecting to node fnalu will actually get you logged into fibi01, an IBM AIX system. This node is fairly limited in resources and is not the best Unix environment for novices and casual Unix users to use. Therefore we will change the alias to point to fsui02, a Sun system that already exists in FNALU. This machine is more user friendly and has significantly more resources available for users. **This change will be made on May 5**.

■

The fourth step we are going to take will encompass all Unix systems that the computing division OSS department manages. In this step we will be installing a common **sendmail** executable and configuration file on all systems. This will give all of the systems we manage a common **sendmail** environment which hopefully in turn lead to fewer problems and less confusion when trying to debug problems. We are in the process of obtaining a common sendmail binary for all the flavors of Unix we support and then we will begin testing a common configuration. We hope to be able to set up all of the systems so they relay all mail to fnal.gov for processing.

There are also steps being taken within other departments in the Computing Division to provide a site IMAP server. Once this system is implemented and available, users will be able to have all of their mail sent to this system where it will be available from practically any machine on or off site. Two major developments lead us to this conclusion and a vigorous interest in the IMAP technology:

1. Available mail readers (pine on Unix and several PC/MAC packages) allow users to access the same mail folders/ systems from desktop PC's, Unix and remote connections.
2. At least one major Web browser (Netscape) is incorporating IMAP into its built in mail handler thus offering the potential of 2 common user interfaces on all platforms.

The first two test installations of IMAP client/servers have pointed out deficiencies in the present implementation but we are now embarking on a third with higher priority When the new IMAP server is available, we will strongly recommend that users who need access to mail from many different locations move their mail to the IMAP service.

Lisa Giacchetti, lisa@fnal.gov

Next Generation Farms at Fermilab

The current generation of Unix farms at Fermilab are rapidly approaching the end of their useful life. The workstations were purchased during the years 1991-1992 and represented the most cost-effective computing available at that time. Acquisition of new workstations is being made to upgrade the Unix farms for the purpose of providing large amounts of computing for reconstruction of data being collected at the 1996-1997 fixed-target run, as well as to provide simulation computing for CMS, the Auger project, accelerator calculations and other projects that require massive amounts of CPU.

The CPU capacity of the farm is approximately 8000 "MIPS", where a "MIP" is defined by the performance of a small simulation and reconstruction code (named TINY). At least three factors contributed to the decision to expand the farms. First, the needs of the 1996-1997 fixed-target run were estimated to be at least 15,000 MIP-years. This estimate was based on assumptions about code performance and data volumes which were the best known at the time (early 1996). Our experience has been that these numbers are underestimates - both the CPU time per event and the data volumes are typically larger than estimated. When one factors that in as well as the efficiency of using the farms it is quickly seen that 8000 MIPS is not sufficient for a timely reconstruction of data from the 1996-1997 run.

Second, there are other large CPU needs at Fermilab which the farms can satisfy. Simulations for the CMS experiment, the Auger project, the Recycler Ring being designed for the next collider run, and superconducting magnets all require large amounts of CPU power. Some theory calculations are also best done on the farms because they require large CPU resources. These programs require large integrated CPU but also larger real memory and processor speeds compared to the 300+ nodes.

Finally, the old nodes are becoming more difficult to maintain and reconfigure. The old SGI nodes, being R3000 based, cannot run IRIX 6.x and the IBM nodes would require additional disk space to allow them to run AIX 4.x. The old farms are divided by routers into Ethernet segments and are logically divided into many NIS domains with multiple NFS servers. Upgrading the OS or memory or modifying the node allocation is difficult on the old farms. We are addressing all of these issues with the design of the new farm.

New Capacity

A project to expand the farms by 15,000 or more MIPS was undertaken in the spring of 1996. To avoid a major porting and testing program it was decided that only the four already-supported Unix operating systems would be allowed to compete for the new farms. They are Digital Unix, IBM AIX, SUN Solaris and SGI IRIX. The expansion was spread over two fiscal years with 7,500 MIPS to be purchased in each year. We purchased 7,500 MIPS in 1996 and in 1997, we will purchase an additional 7,500 MIPS. There was also a purchase of the necessary peripherals and network equipment to build a complete farm.

The details of the first farm expansion are:

TABLE 3.

CPU	Number of CPUs	MIPS/ CPU	Total MIPS
SGI R5000 Challenge S	45	114	5130
IBM RS6000/ 43P (133)	22	115	2530
SGI Challenge DM	2	105	210
IBM J40	2	97	194

The SGI Challenge S and IBM 43P computers comprise the worker nodes and the SGI Challenge DM and the IBM J40 the I/O nodes. Each worker node has 64 MB of real memory and 2 GB of local disk. The I/O nodes each have 128 MB of real memory. In addition to the CPU, peripherals and a switch fabric was purchased to complete the farm. A total of 150 GB of disk and 15 8mm tape drives were purchased and were connected to the two I/O nodes. The entire farm is connected to a Catalyst 5000 switch. The switch has been configured with 72 Ethernet ports, 2 fast-Ethernet ports and 1 FDDI port. The performance of the switch is more than sufficient for the types of computing that will be typically performed on the farm.

A drawing of the new farm is shown in Figure 1.

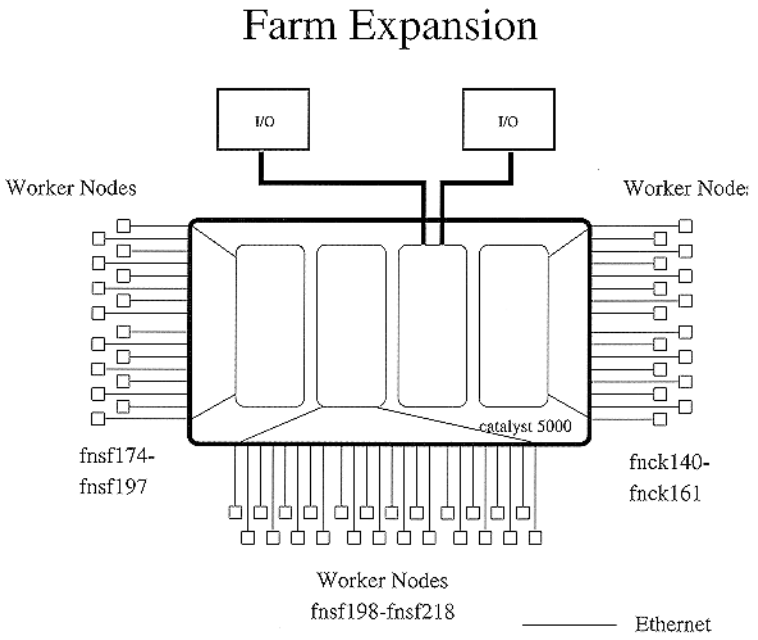

Figure 1

There is much more flexibility built into the farm than was possible in the old farm. All worker and I/O nodes communicate with each other through the switch. This allows rather arbitrary configurations for individual users and jobs. The entire farm is a single NIS domain and file systems are NFS mounted across the farm (AFS is also a possibility). This configuration should allow a much better utilization of the resources of the farm than was possible before.

The second half of the farm expansion is just being completed. The hardware purchased is detailed below. At this time we are unsure whether the IBM 43P's will be 133 MHz or 200 MHz models.

TABLE 4.

CPU	Number of CPUs	MIPS/ CPU	Total MIPS
SGI Origin 200	24	208	4992
IBM RS6000/ 43P (133)	22	115	2530
SGI Origin 200	2	208	416
IBM J40	2	97	194

Software

The software that runs on the farms includes CPS (Cooperative Processes Software), CPS BATCH, and OCS (Operator Console Software). CPS is a toolkit that is used to modify a program and allow it to run across many computers in parallel. The parallelization is normally at the level of an event or collection of events. CPS BATCH is a simple queuing system for jobs. OCS is a program that provides tape drive and tape mount access at the Fermilab Computing Center. CPS has been upgraded to a new version (3.0) which supports 64 bit operating systems. The functionality of the program is left unchanged. CPS BATCH was rewritten to use the same underlying database system as that used by OCS. OCS has been ported to the new systems. The first half of the expansion farm was available for users in early December of 1996.

Future and Summary

During the coming year the second half of the expansion will be integrated into the farms. We expect the fixed-target experiments to finalize their reconstruction code and to start to use the farms heavily. Other large users of CPU will start to use the farm to solve their problems. The challenge for this coming year is to provide a stable system that is used efficiently. It is felt that the faster computers, larger memories, larger disk space, uniform file system, and the switch fabric will all make this easier than in the past.

The farm expansion will increase the total CPU power in the farms to over 20,000 MIPS. This, along with the other facilities at Fermilab, should be sufficient for the near future computing needs. We are starting to think about the computing that will be needed for CDF and D0 during the next collider run, scheduled to begin in 1999. The amount of computing required will be much larger than 20,000 MIPS, meaning that we will be looking at a major increase in computing power near the end of the century. Possible candidates include large SMP's, more Unix farms, and PC farms.

Steve Wolbers, wolbers@fnal.gov

FNALV Downsizing

Within the next few months we will be downsizing the FNALV VAX Cluster. The initial phase of allowing extensions to use the system has ended. In its new configuration, it is to be used only for creating executables for VMS systems and other uses that cannot currently be terminated. If you have continued need for any part of FNALV (login capability, data on disk, data in the archive, etc.) you must contact me even if you previously had an extension.

We are currently in the process of removing disks from FNALV. We are assuming that all data has been retrieved that needs to be retrieved. If this is not the case, let me know immediately. We will begin by recovering disk assigned to project areas, then moving on to personal disk space.

If you have any questions or concerns, please feel free to contact me.

Judith Nicholls, nicholls@fnal.gov

VMS Application Support is Frozen

We are now two years into a nominal three year lab-wide plan for migration away from VMS as a general computing platform. At this time, the Computing Division is formally announcing that VMS application support is frozen.

■

This applies to the software that has traditionally been installed in the LIB:[LIB] directory hierarchy, and to the software supported in the SITE_PRODUCTS structure. It does not apply to operating system software, compilers, or other vendor-supplied code (when applicable) supported by the Operating System Support Department, though support of these is limited in the remaining installations as part of the VMS migration plan.

Budgetary constraints at work inside the Computing Division caused staff reassignments as part of the migration which in turn resulted in insufficient resources to provide software development, testing, or maintenance on VMS.

Our past experience with VMS indicates that non-privileged application code is generally immune to problems resulting from operating system upgrades, compiler or library modifications, etc. We have every reason to believe that this trend will continue. Therefore, many of the products currently in use should continue to function. However, they will not be upgraded by the Computing Division to incorporate new features, bug fixes, or other enhancements.

Dane Skow, dane@fnal.gov, x4730
Lauri Loebel Carpenter, lauri@fnal.gov, x2214

The PC Farm Project

Fermilab has been investigating the use of PC's for HEP computing. As a first step we have built a full offline environment under the Linux operating system on a set of Pentium and Pentium Pro machines (the "PC Farm"). The PC Farm consists of nine PCI-bus personal computers. The hardware configurations of these PC's are purposely varied in order to evaluate the various options available. Six of the systems are Pentium based (P5 166 MHz), and three are Pentium Pro based (P6 200 MHz). One of the Pentium Pro machines is an "SMP" system which has two processors. Seven of the machines use SCSI disk interfaces, and two use EIDE interfaces. The machines are interconnected via a fast Ethernet network. Six of the machines were assembled from components, and three assembled systems were purchased from vendors (Dell, Micron, ASA).

The PC Farm allows us to explore the use of low cost computing hardware for HEP computing problems. Issues that are under investigation include operating systems, utilities and software products, and the possibility of building and testing larger systems for full offline event reconstruction, including parallel processing.

To date, our work has used the Linux operating system. Most of the Linux system infrastructure outside of the kernel (libraries, compilers, utilities) comes from the Free Software Foundation (GNU). Codes written in the C language are compiled using GNU CC (gcc), and Fortran codes are compiled using the g77, Microway, or Absoft compilers, or the f2c converter. In the future we hope to also investigate one or more of the FreeBSD, Solaris, and Microsoft Windows NT operating systems.

Many of the software packages that physicists use as part of everyday work are available for Linux. These include TeX, editors (NEDIT, EMACS), CERNLIB, and so forth. Various pieces of the Fermilab UNIX environment have been ported as well, including UPS (a configuration and product management system) and UPD (a product distribution system). We have ported CPS (Cooperative Processes Software) and have built PVM (Parallel Virtual Machine Package), both toolkits for distributing computational tasks across multiple processors and nodes. Porting of software to Linux has in general been straightforward, and no major problems have been encountered.

The physics codes that we have ported to the PC Farm include the Pythia and VECBOS Monte Carlos, the full CDF offline code (including YBOS), and the GEANT and CDF simulation programs. The successful porting of these codes goes a long ways towards proving the feasibility of using PC's for computing farms.

We have run the Pythia Monte Carlo on all nine nodes simultaneously using CPS, with two important results. First, by comparing execution times to other systems at Fermilab we have confirmed that the P5 and P6 processors run physics codes with efficiencies consistent with major computing benchmarks (e.g., SPECint). On the Pythia Monte Carlo, the 166 MHz Pentium processors are roughly equivalent to 200 MHz SGI R4400 processors; the 200 MHz Pentium Pro processors are roughly 1.4 times faster (113 MIPS vs. 83 MIPS). Second, the two processor SMP machine produced results at twice the rate of the uniprocessor Pentium Pro systems - that is, SMP Linux exhibits 100% scaling on this compute-bound task. Dual, and perhaps quad, processor systems can thus be more cost-effective than uniprocessor systems based upon current prices.

The CDF Production offline reconstruction program was run with CDF Run I data on the PC Farm. Results matching those from other FNAL Unix systems were obtained. Further, as with the Pythia Monte Carlo, the 200 MHz Pentium Pro processors were again observed to be 1.4 times faster than SGI R4400 processors, and the dual processor machine again produced results at twice the rate of the uniprocessor machines.

The PC Farm at present has no tape facilities and only limited disk storage. We hope to verify in the next months that such I/O capabilities can be included and successfully utilized on a cluster of PC's.

■

Parties interested in building and testing software on the PC Farm are encouraged to request accounts by sending mail to **pcfarms@fnal.gov**. For more information, see the PC Farm web page at http://www-ols.fnal.gov/ols/doc/pcfarms/.

Don Holmgren, djholm@fnal.gov, for the PC Farm Team

Status of SDRC IDEAS CAD SOFTWARE

The Mechanical Engineering Design Groups of the laboratory are currently using IDEAS Master Series 2.x. The IDEAS CAD community is planning to move to the current version (IDEAS 4.0) of IDEAS as soon as all user groups have the appropriate hardware (IDEAS 4.0 requires that SGI's systems have at least an R4000 CPU). Since all the Lab groups need to utilize the same version for compatibility, we must resolve the hardware issue before upgrading. Several meetings of the CAD Users Group have been held to identify groups needing equipment upgrades and a plan is in the works to alleviate this situation. The groups that currently use IDEAS are ADCS (Beams), ADMS (Beams), TSS, D0MS (PPD), RDMD (PPD), ADCHL (Beams) and CDF. IDEAS 4.0 is being released for selected TEST installations only at this time.

For more information, please visit the CAD web page at http://www-cad.fnal.gov There also is a CAD USERS mailing list. To subscribe send mail to **mailserv@fnal** with the following text in the body of the message **subscribe caduser**

See http://www-cad.fnal.gov/groupinfo/sdrc/CADUSER_LIST_SERVER.html for more info on the Cad user mailing list. Note that the **caduser@fnal.gov** mailing list is not limited to IDEAS. It also covers Anvil and Ansys, the 2 other supported CAD products at the LAB. There are also periodic CAD User meetings which are announced on the caduser mailing list.

Connie Sieh, OSS Field Systems Group Leader, csieh@fnal.gov
Margaret Greaney, CAD Support, greaney@fnal.gov

C/C++ Error Detection Tool

Insure++ is ParaSoft's tool for detecting programming and runtime errors, memory corruption and memory leaks. It detects compile time errors, runtime errors and third party errors from other packages e.g. X, Motif, curses, and the Unix system calls. Also included with the Insure++ tool are a code coverage analyzer to know what code has and has not been tested and a dynamic memory usage analysis tool providing a visual display of memory usage, and detection of memory hogs. Insure++ was evaluated by using the tool on different components of the DART Data Acquisition Software system and also in collaboration with the Run II Software Engineering Working Group's evaluation of tools for CDF, D0, and the Computing Division.

Features of Insure++ include:

- compile time error detection - incompatible variable declarations, mismatched argument types or function declarations, unused variables and unreached code
- memory related error detection - memory corruption in all memory segments, uninitialized memory reads, dynamic memory errors e.g. double freeing or using memory after its freed, operations on uninitialized, NULL or "wild" pointers, operations on pointers to unrelated data blocks, string manipulation errors, and memory leaks
- C++ specific errors - mixing between new, delete, malloc and free, mismatch between calling **new[]** and **delete** (instead of **delete[]**), virtual function errors and detection of dead code.

Experience with Insure++ and DART

Insure++ analysis was performed on the current version of code for **dis** (Distributed Information Services) and **damp** (Data Acquisition Monitoring). In the case of **dis**, analysis was also done on a previous version known to have leaks and corruption problems.

The compile time and run time error checking of Insure++ proved true. Within moments of the first compilation, coding errors were found that would have been trouble later. At run time, all memory leak points were flagged at the point of exiting the scope where memory was allocated but not freed or at the point of reassigning a pointer value (e.g. with a **strdup**) - marking the code lines where memory allocation took place. Memory boundaries which were compromised by **sprintfs**, **strcats**, array overwrites, etc., were flagged noting the size of the memory written to and the number of bytes written to it which violated boundaries.

In addition, Insure++ saved debugging time by uncovering a problem in the **damp** code which caused a repetitive crash at one of the Dart experiments. The offending code double freed memory already freed within the realloc library routine.

As stated earlier, a version of **dis** known to have several bugs was also analyzed with Insure++. All problems known to exist in the code were found with the exception of one involving corruption of memory allocated and accessed with xdr routines.

Distribution of Insure++

Tar files of the Insure++ product for various operating systems are available for downloading from ParaSoft's web site, http://www.parasoft.com/. Use of the product does require a license from Parasoft.

Instead of providing a full distribution of Insure++ in the kits database, a binding form of the product is available, which includes a ups directory of setup scripts to define the operating environment, a bin directory with a script for downloading the actual body of the product from the vendor web site, and a README file of installation instructions.

To install Insure++ on a node, the Fermi binding product should be distributed first followed by executing the installation instructions provided to download Insure from the vendor site, configure and install it.

Current Availability

Insure++ is available to try on four SGI, IRIX 5.3 nodes on the Fermilab site: CD/OLS nodes fndaub and fndauc, CDF node cdfsga, and DZero node d0chb. Man pages and reference manuals are also available.

For more information contact me or the Software Engineering Working Group (**`sweng@fnal.gov`**).

Carmenita Moore, CD/OLS, carmenita@fnal.gov

Salvaging Data on Overwritten 8mm Magnetic Tapes

Data from 8mm tapes that have been overwritten may be restored. After the last piece of data is written to tape, the Exabyte drive writes a special end of data (EOD) mark. When tapes are overwritten, they then contain two EOD marks: one at the original end of data and one at the end of the new data. Historically, it has been impossible to skip past the new EOD mark and into the salvageable data area.

However, Exabyte does have firmware that will allow specific SCSI commands (e.g., "locate") to skip past the first EOD mark to a specified logical block number. Software support has been integrated into Fermi Tape Tools (**ftt**) starting at version v1_8. Refer to the **ftt** user's guide for detailed instructions.

The Computing Division will provide recovery on a best effort basis for overwritten 8mm tapes until the end of the fixed target run processing.

Note that this capability is only available for 8mm technology. We can not perform data recovery on DLTs.

Margaret Votava, votava@fnal.gov x2625
Jim Meadows, meadows@fnal.gov x4063
Gene Oleynik, DAS Group, oleynik@fnal.gov, x2430 Page: 847-536-1785

NT Driver for CAMAC

The Online Systems Department is releasing a Beta test version of the Windows NT driver for the Jorway Model 411S SCSI Bus CAMAC Highway Driver and the Jorway Model 73A SCSI Bus CAMAC Crate Controller. Both devices are supported by the SJY v1_1 product.

The SJY product supplies a library and dynamically loadable library (dll) for the IEEE CAMAC functions, as well as several program examples. This release supports multi-user configurations. Multi-branch support will be added in a future release if there is sufficient demand. Currently, LAMs can be polled, but not interrupt the PC.

Performance tests with the Jorway 411S showed a DMA startup rate of 0.8ms and data transfer rates of 950Kb/s (parallel CAMAC) and 370Kb/s (serial CAMAC) in 16 bit mode. Both serial and parallel CAMAC had slightly higher performance in 24 bit mode.

The SJY driver is available from kits as product **sjy**, version v1.1, flavor WIN.

Dave Slimmer, slimmer@fnal.gov
Jonathan Streets, streets@fnal.gov